CHAPTER 7

Contents Previous Next

CHAPTER 7 *CDMS Utilities*

7.1 cdscan: Importing datasets into CDMS

7.1.1 Overview

A dataset is a partitioned collection of files. To create a dataset, the files must be scanned to produce a text representation of the dataset. CDMS represents datasets as an ASCII metafile in the CDML markup language. The file contains all metadata, together with information describing how the dataset is partitioned into files. (Note: CDMS provides a direct interface to individual files as well. It is not necessary to scan an individual file in order to access it.)

For CDMS applications to work correctly, it is important that the CDML metafile be valid. The **cdscan** utility generates a metafile from a collection of data files.

CDMS assumes that there is some regularity in how datasets are partitioned:

- A variable can be partitioned (split across files) in at most two dimensions. The partitioned dimension(s) must be either time or vertical level dimensions; variables may not be partitioned across longitude or latitude. Datasets can be parti−tioned by variable as well. For example, one set of files might contain heat fluxes, while another set contains wind speeds.

Otherwise, there is considerable flexibility in how a dataset can be partitioned:

- Files can contain a single variable or all variables in the dataset.
- The time axis can have gaps.
- Horizontal grid boundary information and related information can be duplicated across files.
- Variables can be on different grids.
- Files may be in any of the self−describing formats supported by CDMS, including netCDF, HDF, GrADS/GRIB, and DRS.

7.1.2 cdscan Syntax

The syntax of the cdscan command is

cdscan [*options*] *file1 file2 ...*

or

cdscan [options] **–f** *file_list*

where

- *file1 file2* .. is a blank−separated list of files to scan
- *file_list* is the name of a file containing a list of files to scan, one pathname per line.

Output is written to standard output by default. Use the −x option to specify an output filename.

Table 7.1 cdscan command options

Option Description

−a alias_file	Change variable names to the aliases defined in an alias file. Each line of the alias file consists of two blank separated fields: variable_id alias. variable_id is the ID of the variable in the file, and alias is the name that will be substituted for it in the output dataset. Only variables with entries in the alias_file are renamed.
−c calendar	Specify the dataset calendar attribute. One of "gregorian" (default), "julian", "noleap", "proleptic_gregorian", "standard", or "360_day".
−d dataset_id	String identifier of the dataset. Should not contain blanks or non−printing characters. Default: "**none**"
−e newattr	Add or modify attributes of a file, variable, or axis. The form of newattr is either: var.attr = value to modify a variable or attribute, or .attr = value to modify a global (file) attribute. In either case, value may be quoted to preserve spaces or force the attribute to be treated as a string. If value is not quoted and the first character is a digit, it is converted to integer or floating−point. This option does not modify the input datafiles. See notes and examples below.
−−exclude var,var,...	Exclude specified variables. The argument is a comma−separated list of variables containing no blanks. Also see −−include.
−f file_list	File containing a list of absolute data file names, one per line.
−h	Print a help message.
−i time_delta	Causes the time dimension to be represented as linear, producing a more compact representation. This is useful if the time dimension is very long. *time_delta* is a float or integer. For example, if the time delta is 6 hours, and the reference units are 'hours since xxxx' , set the time delta to 6. See the −r option. See Note 2.
−−include var,var,...	Only include specified variables in the output. The argument is a comma−separated list of variables containing no blanks. Also see −−exclude.
−j	scan time as a vector dimension. Time values are listed individually. Turns off the −i option.
−l levels	Specify that the files are partitioned by vertical level. That is, data for different vertical levels may appear in different files. *levels* is a comma−separated list of levels containing no blanks. See Note 3.
−m levelid	name of the vertical level dimension. The default is the vertical dimension as determined by CDMS. See Note 3.

−p template	Add a file template string, for compatibility with pre−V3.0 datasets. 'cdimport −h' describes template strings.
−q	Quiet mode.
−r time_units	time units of the form "units since yyyy−mm−dd hh:mi:ss", where units is one of "year", "month","day", "hour", "minute", "second".
−s suffix_file	Append a suffix to variable names, depending on the directory containing the data file. This can be used to distinguish variables having the same name but generated by different models or ensemble runs. 'suffix_file' is the name of a file describing a mapping between directories and suffixes. Each line consists of two blank−separated fields: directory suffix. Each file path is compared to the directories in the suffix file. If the file path is in that directory or a subdirectory, the corresponding suffix is appended to the variable IDs in the file. If more than one such directory is found, the first directory found is used. If no match is made, the variable ids are not altered. Regular expressions can be used: see the example in the Notes section.
−t timeid	id of the partitioned time dimension. The default is the name of the time dimension as determined by CDMS. See Note 1.
−−time−linear tzero,delta,units[,calendar]	Override the time dimensions(s) with a linear time dimension. The arguments are comma−separated list: * zero is the initial time point, a floating−point value. * delta is the time delta, floating−point. * units are time units as specified in the [−r] option. * calendar is optional, and is specified as in the [−c] option. If omitted, it defaults to the value specified by [−c], otherwise as specified in the file. Example: −−time−linear '0,1,months since 1980,noleap'
−x xmlfile	Output file name. By default, output is written to standard output.

Notes:

1. Files can be in netCDF, GrADS/GRIB, HDF, or DRS format, and can be listed in any order. Most commonly, the files are the result of a single experiment, and the 'partitioned' dimension is time. The time dimension of a variable is the coordinate variable having a name that starts with 'time' or having an attribute axis='T'. If this is not the case, specify the time dimension with the −t option. The time dimension should be in the form supported by cdtime. If this is not the case (or to override them) use the −r option.
2. By default, the time values are listed explicitly in the output XML. This can cause a problem if the time dimension is very long, say for 6−hourly data. To handle this the form 'cdscan −i delta <files>' may be used. This generates a compact time representation of the form <start, length, delta>. An exception is raised if the time dimension for a given file is not linear.
3. Another form of the command is 'cdscan −l lev1,lev2,..,levn <files>'. This asserts that the dataset is partitioned in both time and vertical level dimensions. The level dimension of a variable is the dimension having a name that starts with "lev", or having an attribute "axis=Z". If this is not the case, set the level name with the −m option.
4. An example of a suffix file:

/exp/pr/ncar−a _ncar−a
/exp/pr/ecm−a _ecm−a
/exp/ta/ncar−a _ncar−a

```
/exp/ta/ecm−a _ecm−a
```

For all files in directory /exp/pr/ncar−a or a subdirectory, the corresponding variable ids will be appended with the suffix '_ncar−a'. Regular expressions can be used, as defined in the Python 're' module. For example, The previous example can be replaced with the single line:

```
/exp/[^/]*/([^/]*) _\g<1>
```

Note the use of parentheses to delimit a group. The syntax \g<n> refers to the nth group matched in the regular expression, with the first group being n=1. The string [^/]* matches any sequence of characters other than a forward slash.

5. Adding or modifying attributes with the −e option:

```
time.units = "days since 1979−1−1"
```

sets the units of all variables/axes to "days since 1979−1−1". Note that since this is done before any other processing is done, it allows overriding of non−COARDS time units.

```
.newattr=newvalue
```

cdscan: Importing datasets into CDMS

Set the global file attribute 'newattr' to 'newvalue'.

6. The [−−time−linear] option overrides the time values in the file(s). The resulting dimension does not have any gaps. In contrast, the [−i], [−r] options use the specified time units (from [−r]), and calendar from [−c] if specified, to convert the file times to the new units. The resulting linear dimension may have gaps.

In either case, the files are ordered by the time values in the files.

The [−−time−linear] option should be used with caution, as it is applied to all the

time dimensions found.

7.1.3 Examples

```
cdscan −c noleap −d test −x test.xml [uv]*.nc
cdscan −d pcmdi_6h −i 0.25 −r 'days since 1979−1−1' *6h*.ctl
```

7.1.4 File Formats

Data may be represented in a variety of self−describing binary file formats, including

- netCDF, the Unidata Network Common Data Format
- HDF, the NCSA Hierarchical Data Format
- GrADS/GRIB, WMO GRIB plus a GrADS control file (.ctl) The first non−comment line of the control file must be a **dset** specification.
- DRS, the PCMDI legacy format.

7.1.5 Name Aliasing

A problem can occur if variables in different files are defined on different grids. What if the axis names are the same? CDMS requires that within a dataset, axis and variable IDs (names) be unique. What should the longitude axes be named in CDMS to ensure uniqueness? The answer is to allow CDMS IDs to differ from file names.

If a variable or axis has a CDMS ID which differs from its name in the file, it is said to have an *alias*. The actual name of the object in the file is stored in the attribute **name_in_file**. **cdscan** uses this mechanism (with the −a and s options) to resolve name conflicts; a new axis or variable ID is generated, and the **name_in_file** is set to the axis name in the file.

Name aliases also can be used to enforce naming standards. For data received from an outside organization, variable names may not be recognized by existing applications. Often it is simpler and safer to add an alias to the metafile rather than rewrite the data

Contents Previous Next